Vietnam J Comput Sci (2017) 4:13–21
DOI 10.1007/s40595-016-0068-y

REGULAR PAPER

Fast support vector clustering

Tung Pham[1] · **Hang Dang**[1] · **Trung Le**[2] · **Thai Hoang Le**[1]

Received: 30 November 2015 / Accepted: 15 April 2016 / Published online: 12 May 2016

Abstract Support-based clustering has recently absorbed plenty of attention because of its applications in solving the difficult and diverse clustering or outlier detection problem. Support-based clustering method perambulates two phases: finding the domain of novelty and performing the clustering assignment. To find the domain of novelty, the training time given by the current solvers is typically over-quadratic in the training size. This fact impedes the application of support-based clustering method to the large-scale datasets. In this paper, we propose applying stochastic gradient descent framework to the first phase of support-based clustering for finding the domain of novelty in the form of a half-space and a new strategy to perform the clustering assignment. We validate our proposed method on several well-known datasets for clustering task to show that the proposed method renders a comparable clustering quality to the baselines while being faster than them.

Keywords Support vector clustering · Cluster analysis · Kernel method

1 Introduction

Cluster analysis is a fundamental problem in pattern recognition where objects are categorized into groups or clusters based on pairwise similarities between those objects such that two criteria, homogeneity and separation, are achieved [21]. Two challenges in the task of cluster analysis are (1) dealing with complicated data with nested or hierarchy structures inside; and (2) automatically detecting the number of clusters. Recently, support-based clustering, e.g., support vector clustering (SVC) [1], has drawn a significant research concern because of its applications in solving the difficult and diverse clustering or outlier detection problem [1,2,8,10,11,15,23]. These clustering methods have two main advantages comparing with other clustering methods: (1) ability to generate the clustering boundaries with arbitrary shapes and automatically discover the number of clusters; and (2) capability to handle well the outliers.

Support-based clustering methods always undergo two phases. In the first phase, the domain of novelty, e.g., optimal hypersphere [1,9,22] or hyperplane [18], is discovered in the feature space. The domain of novelty when mapped back into the input space will become a set of contours tightly enclosing data which can be interpreted as cluster boundaries. However, this set of contours does not specify how to assign a data sample to its cluster. In addition, the computational complexity of the current solvers [3,7] to find out the domain of novelty is often over-quadratic [4]. Such a computational complexity impedes the usage of support-based clustering methods for the real-world datasets. In the second phase, namely clustering assignment, based on the geometry information carried in the resultant set of contours harvested from the first phase, data samples are appointed to their clusters. Several works have been proposed for improving cluster assignment procedure [2,8,11,15,23].

Recently, stochastic gradient descent (SGD) frameworks [6,19,20] have emerged as building blocks to develop the learning methods for efficiently handling the large-scale dataset. SGD-based algorithm has the following advantages: (1) very fast; (2) ability to run in online mode; and (3)

✉ Hang Dang
dthang@hcmus.edu.vn

[1] Faculty of Information Technology, VNUHCM-University of Science, Ho Chi Minh City, Vietnam

[2] Faculty of Information Technology, HCMc University of Pedagogy, Ho Chi Minh City, Vietnam

not requiring to load the entire dataset to the main memory in training. In this paper, we conjoin the advantages of SGD with support-based clustering. In particular, we propose to use the optimal hyperplane as the domain of novelty. The margin, i.e., the distance from the origin to the optimal hyperplane, is maximized to make the contours enclosing the data as tightly as possible. We subsequently apply the stochastic gradient descent framework proposed in [19] to the first phase of support-based clustering for achieving the domain of novelty. Finally, we propose a new strategy for clustering assignment where each data sample in the extended decision boundary has its own trajectory to converge to an equilibrium point and clustering assignment is then reduced to the same task for those equilibrium points. Our clustering assignment strategy distinguishes from the existing works of [8,11–13] in the way to find the trajectory with a start and the initial set of data samples that need to do a trajectory for finding the corresponding equilibrium point. The experiments established on the real-world datasets show that our proposed method produces the comparable clustering quality with other support-based clustering methods while simultaneously achieving the computational speedup.

To summarize, the contribution of the paper consists of the following points:

- Different from the works of [1,2,11,15,23] which employ a hypersphere to characterize the domain of novelty, we propose using a hyperplane to characterize the domain of novelty. This allows us to introduce SGD-based solution for finding the domain of novelty.
- We propose SGD-based solution for finding the domain of novelty. We perform a rigorous convergence analysis for the proposed solution. We note that the works of [1,2,11,15,23] utilized the Sequential-Minimal-Optimization-based approach [17] to find the domain of novelty wherein the computational complexity is over-quadratic and it requires loading the entire Gram matrix to the main memory.
- We propose new clustering assignment strategy which can reduce the clustering assignment for N samples in the entire training set to the same task for M equilibrium points where M is usually very small comparing with N.
- Comparing with the conference version [16], this paper presents a more rigorous convergence analysis with the full proofs and explanations. In addition, it further introduces new strategy for clustering assignment. Regarding the experiment, it compares with more baselines and produces more experimental results.

2 Stochastic gradient descent large margin one-class support vector machine

2.1 Large margin one-class support vector machine

Given the dataset $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$, to define the domain of novelty, we construct an optimal hyperplane that can separate the data samples and the origin such that the margin, i.e., the distance from the origin to the hyperplane, is maximized. The optimization problem is formulated as

$$\max_{\mathbf{w},\rho} \left(\frac{|\rho|}{\|\mathbf{w}\|^2} \right)$$

subjects to

$$\mathbf{w}^\mathsf{T}\phi(x_i) - \rho \geq 0, \quad i = 1, \ldots, N$$
$$\mathbf{w}^\mathsf{T}\mathbf{0} - \rho = -\rho < 0$$

where ϕ is a transformation from the input space to the feature space and $\mathbf{w}^\mathsf{T}\phi(x) - \rho = 0$ is equation of the hyperplane. It occurs that the margin is invariant if we scale $(\mathbf{w}, \rho)$ by a factor k. Hence without loss of generality, we can assume that $\rho = 1$ and we achieve the following optimization problem

$$\min_{\mathbf{w}} \left(\frac{1}{2}\|\mathbf{w}\|^2 \right)$$

subjects to

$$\mathbf{w}^\mathsf{T}\phi(x_i) - 1 \geq 0, \quad i = 1, \ldots, N$$

Using the slack variables, we can extend the above optimization problem to form the soft model of large margin one-class Support vector machine (LMOCSVM)

$$\min_{\mathbf{w}} \left(\frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{N}\sum_{i=1}^{N}\xi_i \right)$$

subjects to

$$\mathbf{w}^\mathsf{T}\phi(x_i) - 1 \geq -\xi_i, \quad i = 1, \ldots, N$$
$$\xi_i \geq 0, \quad i = 1, \ldots, N$$

where $C > 0$ is the trade-off parameter and $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_N]$ is the vector of slack variables.

We can rewrite the above optimization problem in the primal form as follows

$$\min_{\mathbf{w}}\left(J(\mathbf{w}) \triangleq \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{N}\sum_{i=1}^{N}\max\left\{0, 1-\mathbf{w}^{\mathsf{T}}\phi(x_i)\right\}\right) \tag{1}$$

2.2 SGD-based Solution in the primal form

To efficiently solve the optimization in Eq. (1), we use stochastic gradient descent method. We name the outcome method by stochastic-based large margin one-class support vector machine (SGD-LMSVC).

At tth round, we sample the data point x_{n_t} from the dataset $\mathcal{D}$. Let us define the instantaneous function $g_t(\mathbf{w}) \triangleq \frac{1}{2}\|\mathbf{w}\|^2 + C\max\left\{0, 1-\mathbf{w}^{\mathsf{T}}\phi\left(x_{n_t}\right)\right\}$. It is obvious that $g_t(\mathbf{w})$ is 1 − strongly convex w.r.t the norm $\|.\|_2$ over the feature space.

The learning rate is $\eta_t = \frac{1}{t}$ and the sub-gradient is $\lambda_t = \mathbf{w}_t - C\mathbf{I}_{[\mathbf{w}_t^{\mathsf{T}}\phi(x_{n_t})<1]}\phi\left(x_{n_t}\right) \in \partial g_t(\mathbf{w}_t)$, where $\mathbf{I}_A(.)$ is the indicator function. Therefore, the update rule is

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t\lambda_t = \left(1-\frac{1}{t}\right)\mathbf{w}_t + \frac{C}{t}\mathbf{I}_{[\mathbf{w}_t^{\mathsf{T}}\phi(x_{n_t})<1]}\phi\left(x_{n_t}\right) \tag{2}$$

Algorithm 1 Algorithm for solving SGD-LMSVC in the primal form.

Input: C, $K(.,.)$, $\mathcal{D} = \{x_1, ..., x_N\}$
$\mathbf{w}_1 = \mathbf{0}$
for $t = 1$ **to** T **do**
  Sampling n_t from $[N] = \{1, 2, ..., N\}$.
  $\mathbf{w}_{t+1} = \left(1-\frac{1}{t}\right)\mathbf{w}_t + \frac{C}{t}\mathbf{I}_{[\mathbf{w}_t^{\mathsf{T}}\phi(x_{n_t})<1]}\phi\left(x_{n_t}\right)$.
endfor
Output: $\mathbf{w}_{T+1}$

Algorithm 1 is proposed to find the optimal hyperplane which defines the domain of novelty. At each round, one data sample is uniformly sampled from the training set and the update rule in Eq. (2) is applied to determine the next hyperplane, i.e., $\mathbf{w}_{t+1}$. Finally, the last hyperplane, i.e., $\mathbf{w}_{T+1}$ is outputted as the optimal hyperplane. According to the theory displayed in the next section, we can randomly output any intermediate hyperplane and the approximately accurate solution is still warranted in a long-term training. Nonetheless, in Algorithm 1, we make use of the last hyperplane as output to exploit as much as possible the information accumulated through the iterations. It is worthwhile to note that in Algorithm 1, we store $\mathbf{w}_t$ as $\mathbf{w}_t = \sum_i \alpha_i\phi(x_i)$.

2.3 Convergence analysis

In this section, we show the convergence analysis of Algorithm 1. We assume that data are bounded in the feature space, that is, $\|\phi(x)\| \leq R, \ \forall x \in \mathcal{X}$. We denote the optimal solution by $\mathbf{w}^*$, that is, $\mathbf{w}^* = \text{argmin}_{\mathbf{w}}\mathcal{J}(\mathbf{w})$. We derive as follows.

Lemma 1 establishes a bound on $\|\mathbf{w}_T\|$, followed by Lemma 2 which establishes a bound on $\|\lambda_T\|$.

Lemma 1 *The following statement holds*

$$\|\mathbf{w}_T\| \leq CR, \quad \forall T$$

Proof We have

$$t\mathbf{w}_{t+1} = (t-1)\mathbf{w}_t + C\mathbf{I}_{[\mathbf{w}_t^{\mathsf{T}}\phi(x_{n_t})<1]}\phi\left(x_{n_t}\right)$$
$$t\|\mathbf{w}_{t+1}\| \leq (t-1)\|\mathbf{w}_t\| + CR$$

Taking sum the above when $t = 1, 2, \ldots, T-1$, we gain

$$(T-1)\|\mathbf{w}_T\| \leq (T-1)CR$$
$$\|\mathbf{w}_T\| \leq CR$$

Lemma 2 *The following statement holds*

$$\|\lambda_T\| = \left\|\mathbf{w}_t - C\mathbf{I}_{[\mathbf{w}_t^{\mathsf{T}}\phi(x_{n_t})<1]}\phi\left(x_{n_t}\right)\right\| \leq 2CR, \quad \forall T$$

Proof We have

$$\|\lambda_T\| \leq \|\mathbf{w}_T\| + CR \leq 2CR$$

Theorem 1 establishes a bound on regret and shows that Algorithm 1 has the convergence rate $O\left(\frac{\log T}{T}\right)$.

Theorem 1 *Let us consider the running of Algorithm 1.The following statement holds*

$$\mathcal{J}(\overline{\mathbf{w}}_T) - \mathcal{J}\left(\mathbf{w}^*\right) \leq \frac{2C^2R^2(\log T + 1)}{T}$$

where $\overline{\mathbf{w}}_T = \frac{1}{T}\sum_{t=1}^{T}\mathbf{w}_t$.

Proof It is apparent that

$$\mathbb{E}[\lambda_t|\mathbf{w}_t] = \mathbf{w}_t - \frac{C}{N}\sum_{n_t=1}^{N}C\mathbf{I}_{[\mathbf{w}_t^{\mathsf{T}}\phi(x_{n_t})<1]}\phi\left(x_{n_t}\right) = \mathcal{J}'(\mathbf{w}_t)$$

We have the following

$$\begin{aligned}\left\|\mathbf{w}_{t+1} - \mathbf{w}^*\right\|^2 &= \left\|\mathbf{w}_t - \eta_t\lambda_t - \mathbf{w}^*\right\|^2 \\ &\leq \left\|\mathbf{w}_t - \mathbf{w}^*\right\|^2 - 2\eta_t\lambda_t^{\mathsf{T}}\left(\mathbf{w}_t - \mathbf{w}^*\right) \\ &\quad + \eta_t^2\|\lambda_t\|^2\end{aligned}$$

Taking conditional expectation w.r.t $\mathbf{w}_t$ the above, we gain

$$\begin{aligned}
&\mathcal{J}^{'}(\mathbf{w}_t)\left(\mathbf{w}_t-\mathbf{w}^*\right)\\
&\leq \frac{\mathbb{E}\left[\|\mathbf{w}_t-\mathbf{w}^*\|^2\right]-\mathbb{E}\left[\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2\right]}{2\eta_t}+\frac{\eta_t}{2}\mathbb{E}\left[\|\lambda_t\|^2\right]\\
&\mathcal{J}(\mathbf{w}_t)-\mathcal{J}\left(\mathbf{w}^*\right)+\frac{1}{2}\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\\
&\leq \frac{\mathbb{E}\left[\|\mathbf{w}_t-\mathbf{w}^*\|^2\right]-\mathbb{E}\left[\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2\right]}{2\eta_t}+\frac{\eta_t}{2}\mathbb{E}\left[\|\lambda_t\|^2\right]\\
&\mathcal{J}(\mathbf{w}_t)-\mathcal{J}\left(\mathbf{w}^*\right)\leq\frac{t-1}{2}\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]\\
&-\frac{t}{2}\mathbb{E}\left[\left\|\mathbf{w}_{t+1}-\mathbf{w}^*\right\|^2\right]+\frac{2C^2R^2}{t}
\end{aligned}$$

Taking expectation again, we achieve

$$\begin{aligned}
\mathbb{E}\left[\mathcal{J}(\mathbf{w}_t)\right]-\mathcal{J}\left(\mathbf{w}^*\right)\leq\frac{t-1}{2}\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]\\
-\frac{t}{2}\mathbb{E}\left[\left\|\mathbf{w}_{t+1}-\mathbf{w}^*\right\|^2\right]+\frac{2C^2R^2}{t}
\end{aligned}$$

Taking sum the above inequality when $t=1,\ldots,T$, we gain

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\mathcal{J}(\mathbf{w}_t)\right]-\mathcal{J}\left(\mathbf{w}^*\right)\\
&\leq\frac{2C^2R^2}{T}\sum_{t=1}^{T}\frac{1}{t}\leq\frac{2C^2R^2(\log\,T+1)}{T} \qquad (3)\\
&\mathcal{J}(\overline{\mathbf{w}}_T)-\mathcal{J}\left(\mathbf{w}^*\right)\leq\frac{2C^2R^2(\log\,T+1)}{T}
\end{aligned}$$

□

Theorem 1 shows the inequality for the average solution in the expectation form. In the following theorem, we prove that if we output a single-point solution, with a high probability we have a real inequality.

Theorem 2 *Let us consider the running of Algorithm 1. Let r be an integer randomly picked from* $\{1,2,\ldots,T\}$. *Given* $\delta\in(0;1)$, *with the probability greater than* $1-\delta$ *the following inequality holds*

$$\mathcal{J}(\mathbf{w}_r)<\mathcal{J}\left(\mathbf{w}^*\right)+\frac{2R^2C^2(1+log\,T)}{\delta T}$$

Proof Let us denote $X=\mathcal{J}(\mathbf{w}_r)-\mathcal{J}(\mathbf{w}^*)\geq 0$. By definition of r, we have

$$\mathbb{E}_r[X]=\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\mathcal{J}(\mathbf{w}_t)\right]-\mathcal{J}\left(\mathbf{w}^*\right)$$

$$\mathbb{E}[X]=\mathbb{E}_{(x_t,y_t)_{t=1}^{T}}\left[\mathbb{E}_r[X]\right]\leq\frac{2C^2R^2(\log\,T+1)}{T}$$

Using Markov inequality, we gain

$$\begin{aligned}
&\mathbb{P}(X\geq\varepsilon)\leq\frac{\mathbb{E}[X]}{\varepsilon}\leq\frac{2C^2R^2(\log\,T+1)}{\varepsilon T}\\
&\mathbb{P}(X<\varepsilon)>1-\frac{2C^2R^2(\log\,T+1)}{\varepsilon T}
\end{aligned}$$

Choosing $\delta=\frac{2C^2R^2(\log\,T+1)}{\varepsilon T}$, we gain the conclusion. □

We now investigate the number of iterations required if we want to gain an ε-precision solution with a probability at least $1-\delta$. According to Theorem 2, the number of iterations T must be greater than T_0 where T_0 is the smallest number such that

$$\begin{aligned}
&\frac{2R^2C^2(1+\log T_0)}{\delta T_0}\leq\varepsilon\\
&\frac{1+\log T_0}{T_0}\leq\frac{\varepsilon\delta}{2R^2C^2}
\end{aligned}$$

3 Clustering assignment

After solving the optimization problem, we yield the decision function

$$f(x)=\sum_{i=1}^{N}\alpha_i K(x_i,x)-1$$

To find the equilibrium points, we need to solve the equation $\nabla f(x)=0$. To this end, we use the fixed point technique and assume that Gaussian kernel is used, i.e., $K\left(x,x'\right)=e^{-\gamma\|x-x'\|^2}$. We then have

$$\begin{aligned}
\frac{1}{2}\nabla f(x)&=\sum_{i=1}^{N}\alpha_i(x_i-x)e^{-\gamma\|x-x_i\|^2}\\
&=0\rightarrow x=\frac{\sum_{i=1}^{N}\alpha_i e^{-\gamma\|x-x_i\|^2}x_i}{\sum_{i=1}^{N}\alpha_i e^{-\gamma\|x-x_i\|^2}}=P(x)
\end{aligned}$$

To find an equilibrium point, we start with the initial point $x^{(0)}\in\mathbb{R}^d$ and iterate $x^{(j+1)}=P\left(x^{(j)}\right)$. By fixed point theorem, the sequence $x^{(j)}$, which can be considered as a trajectory with start $x^{(0)}$, converges to the point $x_*^{(0)}$ satisfying $\mathrm{P}(x_*^{(0)})=x_*^{(0)}$ or $\nabla f(x_*^{(0)})=0$, i.e., $x_*^{(0)}$ is an equilibrium point.

Let us denote $B_\epsilon=\{x_i:1\leq i\leq N\wedge|f(x_i)|\leq\epsilon\}$, namely the extended boundary for a tolerance $\epsilon>0$. It follows that the set B_ϵ forms a strip enclosing the decision boundary $f(x)=0$. Algorithm 2 is proposed to do clustering assignment. In Algorithm 2, the task of clustering assignment is reduced to itself for M equilibrium point. To fulfill cluster assignment for M equilibrium points, we run $m=20$ sample-point test as proposed in [1].

Algorithm 2 Clustering assignment procedure.

Input: $f(x) = \sum_{i=1}^{N} \alpha_i K(x_i, x) - 1$, B_ε, $\mathcal{D} = \{x_1, \ldots, x_N\}$
$E = \emptyset$.
foreach $x^{(0)}$ **in** B_ϵ **do**
  Find the equilibrium point $x_*^{(0)}$.
  if $\left(x_*^{(0)} \notin E\right)$ **then** $E = E \cup \left\{x_*^{(0)}\right\}$
endfor
//Assume that $E = \{e_1, e_2, \ldots, e_M\}$
Do m sample point test with for E to find cluster indices for $e_1, e_2, \ldots, e_M$.
Each point $x^{(0)} \in B_\epsilon$ is assigned to the cluster of its corresponding equilibrium point $x_*^{(0)} \in E$.
Each point $x \in \mathcal{D} \backslash B_\epsilon$ is assigned to the cluster of its nearest neighbor in B_ϵ using the Euclidean distance.
Output: clustering solution for $\mathcal{D} = \{x_1, ..., x_N\}$

Our proposed clustering assignment procedure is different with the existing procedure proposed in [1]. The procedure proposed in [1] requires to run $m = 20$ sample-point test for every edge connected x_i, x_j $(i \neq j)$ in the training set. Consequently, the computational cost incurred is $O(N(N-1)ms)$ where s is the sparsity level of the decision function (i.e., the number of vectors in the model). Our proposed procedure needs to perform $m = 20$ sample-point test for a reduced set of M data samples (i.e., the set of the equilibrium points $\{e_1, e_2, \ldots, e_M\}$) where M is possibly very small comparing with N. The reason is that many data points in the training set could converge to a common equilibrium point which significantly reduces the size from N to M. The computational cost incurred is therefore $O(M(M-1)ms)$.

4 Experiments

4.1 Visual experiment

To visually show the high clustering quality produced by our proposed SGD-LMSVC, we establish experiment on three synthesized datasets and visually make comparison SGD-LMSVC with C-Means and Fuzzy C-Means. In the first experiment, data samples form the nested structure with two outside rings and one Gaussian distribution at center. As shown in Fig. 1, SGD-LMSVC can perfectly detect three clusters without any prior information while both C-Means and Fuzzy C-Means with the number of clusters being set to 3 beforehand fail to discover the nested clusters. The second experiment is carried out with a two-moon dataset. As observed from Fig. 2, SGD-LMSVC without any prior knowledge can flawlessly discover two clusters in moons, however, C-Means and Fuzzy C-Means cannot detect the clusters correctly. In the last visual experiment, we generate data from the mixture of 4 Gaussian distributions. As shown in Fig. 3, SGD-LMSVC can perfectly detect 4 clusters corresponding to the individual Gaussian distributions. These visual experiments manifest that SGD-LMSVC is able to generate the cluster boundaries in arbitrary shapes as well as automatically detect the appropriate number of clusters well presented in the data.

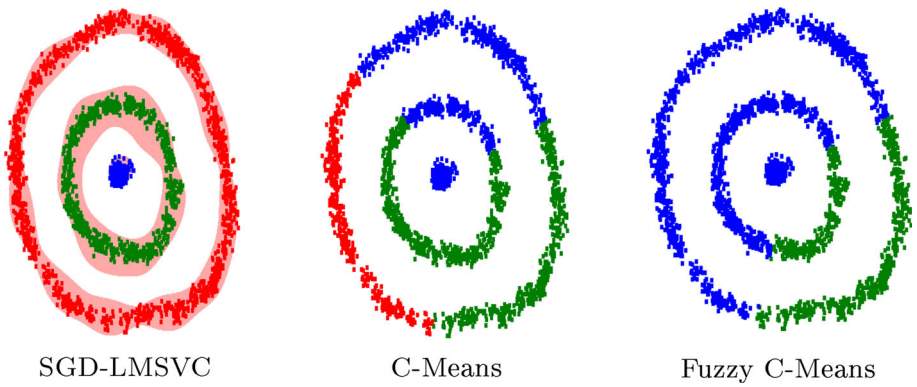

Fig. 1 Visual comparison of SGD-LMSVC (the orange region is the domain of novelty) with C-Means and Fuzzy C-Means on two ring dataset

4.2 Experiment on real datasets

To explicitly prove the performance of the proposed algorithm, we establish experiments on the real datasets. Clustering problem is basically an unsupervised learning task and, therefore, there is not a perfect measure to compare given two clustering algorithms. We examine five typical clustering validity indices (CVI) including compactness, purity, rand index, Davies–Bouldin index (DB index), and normalized mutual information (NMI). A good clustering algorithm should produce a solution which has a high purity, rand index, DB index, and NMI and a low compactness.

4.2.1 Clustering validity index

Compactness measures the average pairwise distances of points in the same cluster [5] and is given as follows

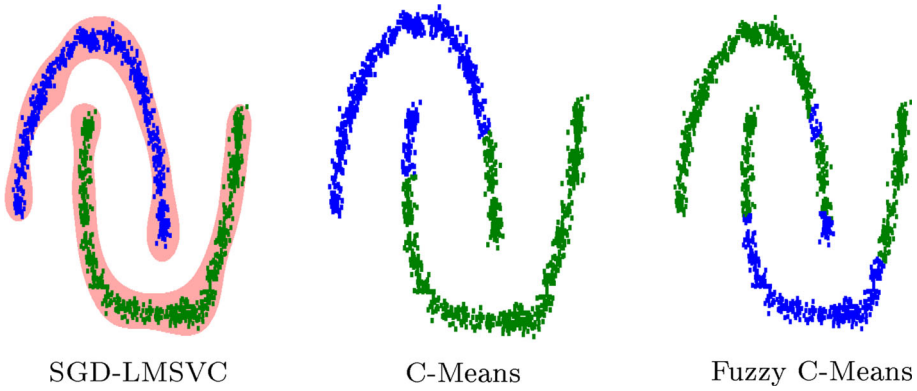

Fig. 2 Visual comparison of SGD-LMSVC (the orange region is the domain of novelty) with C-Means and Fuzzy C-Means on two-moon dataset

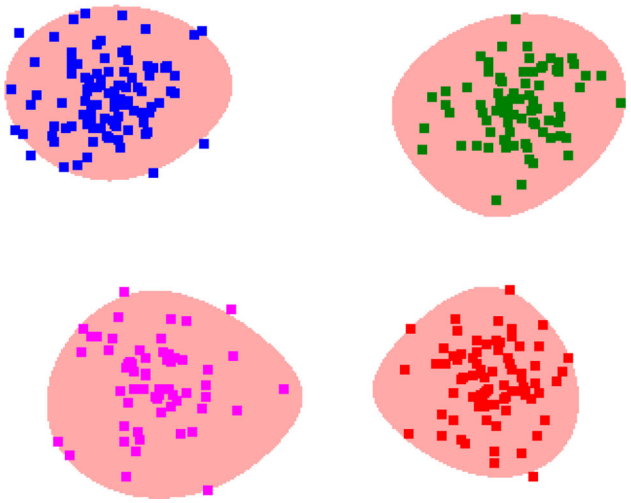

Fig. 3 SGD-LMSVC (the orange region is the domain of novelty) can recognize the clusters scattered from mixture of four Gaussian distributions

$$\text{Compactness} \triangleq \frac{1}{N}\sum_{k=1}^{m} N_k \frac{\sum_{x,x' \in C_k} d\left(x, x'\right)}{N_k\left(N_k - 1\right)/2}$$

where the cluster solution consists of m clusters C_1, C_2, $\ldots$, C_m whose cardinalities are N_1, N_2, $\ldots$, N_m, respectively.

The clustering with a small compactness is preferred. A small compactness gained means the average intra-distance of clusters is small and homogeneity is thereby good, i.e., two objects in the same cluster have high similarity to each other.

The second CVI in use is purity which measures the purity of clustering solution with respect to the nature classes of data [14]. It is certainly true that the metric purity is only appropriate for data with labels in nature. Let N_{ij} be the number of objects in cluster i that belong to the class j. Again, let $N_i \triangleq \sum_{j=1}^{m} N_{ij}$ be total number of objects in cluster i. Let us define $p_{ij} \triangleq \frac{N_{ij}}{N_j}$, i.e., the empirical distribution over class labels for cluster i. We define a purity of a cluster as $p_i \triangleq \max_j p_{ij}$ and overall purity of a clustering solution as

$$\text{Purity} \triangleq \sum_i \frac{N_i}{N} \times p_i$$

The purity ranges between 0 (bad) and 1 (good). This CVI embodies the classification ability of clustering algorithm. A clustering algorithm which achieves a high purity can be appropriately used for classification purpose.

The third CVI used as a measure is rand index [14]. To calculate this CVI for a clustering solution, we need to construct a 2×2 contingency table containing the following numbers: (1) TP (true positive) is the number of pairs that are in the same cluster and belong to the same class; (2) TN (true negative) is the number of pairs that are in two different clusters and belong to different classes; (3) FP (false positive) is the number of pairs that are in the same cluster but belong to different classes; and (4) FN (false negative) is the number of pairs that are in two different clusters but belong to the same class. Rand index is defined as follows

$$\text{Rand} \triangleq \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$$

This can be interpreted as the fraction of clustering decisions that are correct. Obviously, rand index ranges between 0 and 1.

Davies–Bouldin validity index is a function of the ratio of the sum of intra-distances to inter-distances [5] and is formulated as follows

$$\text{DBI} \triangleq \frac{1}{m}\sum_{i=1}^{m} \max_{j \neq i} \left\{ \frac{\Delta\left(C_i\right) + \Delta\left(C_j\right)}{d\left(C_i, C_j\right)} \right\}$$

A good clustering algorithm should produce the solution which has as smallest DBI as possible.

The last considered CVI is normalized mutual information (NMI) [14]. This measure allows us to trade off the quality of the clustering against the number of clusters.

$$\text{NMI} \triangleq \frac{I\left(\Omega, C\right)}{\left[H\left(C\right) + H\left(\Omega\right)\right]/2}$$

where $C = \{c_1, \ldots, c_J\}$ is the set of classes and $\Omega = \{\omega_1, \ldots, \omega_K\}$ is the set of clusters. $I(\Omega, C)$ is the mutual information and is defined as

$$I(\Omega, C) \triangleq \sum_k \sum_j P\left(c_j \bigcap \omega_k\right) \log \frac{P\left(c_j \bigcap \omega_k\right)}{P\left(c_j\right) P\left(\omega_k\right)}$$

and $H(.)$ is the entropy and is defined as

$$H(\Omega) \triangleq -\sum_k P(\omega_k) \log P(\omega_k)$$

Table 1 The statistics of the experimental datasets

Datasets	Size	Dimension	#Classes
Aggregation	788	2	7
Breast cancer	699	9	2
Compound	399	2	6
D31	3100	2	31
Flame	240	2	2
Glass	214	9	7
Iris	150	4	3
Jain	373	2	2
Pathbased	300	2	3
R15	600	2	15
Spiral	312	2	3
Abalone	4177	8	28
Car	1728	6	4
Musk	6598	198	2
Shuttle	43,500	9	5

It is certainly that the NMI ranges between 0 and 1, and a good clustering algorithm should produce as highest NMI measure as possible.

We perform experiments on 15 well-known datasets for clustering task. The statistics of the experimental datasets is given in Table 1. These datasets are fully labeled and consequently, the CVIs like purity, rand index, and NMI can be completely estimated. We make comparison of our proposed SGD-LMSVC with the following baselines.

4.2.2 Baselines

- *Support vector clustering (SVC)* [1] using LIBSVM [3] for finding domain of novelty and fully connected graph for clustering assignment.
- *Fast support vector clustering (FSVC)* [8] an equilibrium-based approach for clustering assignment.

It is noteworthy that the first phase in our proposed SGD-LMSVC is SGD-based solution for LMOCSVM (cf. Algorithm 1) and the second phase is proposed in Algorithm 2. All competitive methods are run on a Windows computer with dual-core CPU 2.6 GHz and 4 GB RAM.

4.2.3 Hyperparameter setting

The RBF kernel, given by $K\left(x, x'\right) = e^{-\gamma \|x - x'\|^2}$, is employed. The width of kernel γ is searched on the grid $\left\{2^{-5}, 2^{-3}, \ldots, 2^3, 2^5\right\}$. The trade-off parameter C is searched on the same grid. In addition, the parameters p and ε in FSVC are searched in the common grid

Table 2 The purity, rand index, and NMI of the clustering methods on the experimental datasets

Datasets	Purity			Rand index			NMI		
	SVC	SGD	FSVC	VC	SGD	FSVC	SVC	SGD	FSVC
Aggregation	**1.00**	**1.00**	0.22	**1.00**	**1.00**	0.22	0.69	**0.75**	0.60
Breast cancer	0.98	**0.99**	**0.99**	0.82	**0.85**	0.81	0.22	**0.55**	0.45
Compound	**0.66**	0.62	0.13	**0.92**	0.88	0.25	0.51	**0.81**	0.45
Flame	0.86	**0.87**	0.03	0.75	**0.76**	0.03	**0.55**	0.51	0.05
Glass	0.5	**0.71**	0.65	0.77	**0.91**	0.54	**0.60**	0.44	0.53
Iris	**1.00**	**1.00**	0.68	**0.97**	0.96	0.69	0.63	**0.75**	0.71
Jain	0.37	0.46	**0.69**	0.7	0.71	**0.77**	0.53	0.31	**1.00**
Pathbased	0.6	0.5	**1.00**	0.81	0.94	**1.00**	**0.48**	0.43	0.12
R15	0.88	**0.9**	0.37	**0.74**	0.71	0.37	0.67	**0.77**	**0.77**
Spiral	0.09	0.33	**0.53**	0.15	**0.94**	0.75	**0.52**	0.34	0.16
D31	0.94	**0.99**	0.42	**0.88**	0.81	0.54	0.45	**0.50**	0.38
Abalone	0.22	**0.44**	0.03	0.43	**0.86**	0.12	0.22	**0.34**	0.07
Car	0.94	**0.95**	0.70	0.46	0.46	**0.54**	**0.32**	**0.32**	0.24
Musk	0.87	0.68	**0.88**	0.26	**0.28**	0.26	0.21	0.16	**0.23**
Shuttle	**0.06**	0.05	**0.06**	**0.84**	0.83	0.75	0.26	0.41	**0.50**

Table 3 The compactness and DB index of the clustering methods on the experimental datasets

Datasets	Compactness			DB index		
	SVC	SGD	FSVC	SVC	SGD	FSVC
Aggregation	**0.29**	**0.29**	2.84	**0.68**	0.67	0.63
Breast cancer	1.26	**0.68**	0.71	**1.58**	1.38	0.53
Compound	0.5	**0.21**	2.43	**2.45**	0.86	0.67
Flame	0.58	**0.44**	2.28	**1.3**	0.65	3.56
Glass	0.72	**0.68**	1.85	0.53	0.56	**0.93**
Iris	0.98	**0.25**	0.99	**1.95**	1.17	0.77
Jain	0.96	**0.36**	1.16	**1.23**	1.08	0.71
Pathbased	**0.18**	0.3	1.04	0.36	0.73	**1.07**
R15	0.61	**0.13**	1.84	**2.96**	1.42	1.37
Spiral	2	**0.17**	0.18	**1.41**	0.98	0.36
D31	1.41	**0.26**	1.78	**2.33**	1.35	1.21
Abalone	3.88	**0.40**	4.97	3.78	**3.91**	1.29
Car	0.75	**0.74**	14.68	**1.76**	**1.76**	1.57
Musk	**9.89**	30.05	20.00	2.27	**2.83**	0.01
Shuttle	0.50	0.46	**0.26**	**1.86**	1.84	1.32

Table 4 Training time in second (i.e., the time for finding domain of novelty) and clustering time in second (i.e., the time for clustering assignment) of the clustering methods on the experimental datasets

Datasets	Training time			Clustering time		
	SVC	SGD	FSVC	SVC	SGD	FSVC
Aggregation	0.05	**0.03**	0.05	31.42	**2.83**	7.51
Breast cancer	0.18	**0.02**	0.05	19.80	**2.14**	22.86
Compound	0.03	**0.02**	0.10	6.82	**1.17**	7.24
Flame	**0.02**	**0.02**	15.16	1.81	**0.67**	4.31
Glass	0.03	0.03	**0.02**	2.30	**0.53**	10.67
Iris	**0.02**	**0.02**	0.04	1.03	**0.34**	4.33
Jain	**0.02**	**0.02**	0.03	5.80	**0.81**	4.59
Pathbased	**0.02**	**0.02**	0.05	4.02	**0.54**	4.22
R15	**0.02**	**0.02**	**0.02**	4.14	**3.68**	10.43
Spiral	**0.02**	**0.03**	**0.02**	1.60	**0.99**	7.78
D31	0.17	**0.09**	**0.09**	467.72	**6.56**	33.08
Abalone	2.26	**0.81**	10.94	653.65	**26.58**	242.97
Car	5.62	**0.64**	8.15	67.66	**7.05**	84.47
Musk	55.93	**5.79**	58.49	602.09	**432.58**	510.25
Shuttle	10.03	**0.46**	68.43	1,972.61	**925**	1,125.46

$\{0.1, 0.2, \ldots, 0.9, 1\}$ which is the same as in [8]. Determining the number of iterations in Algorithm 1 is really challenging. To resolve it, we use the stopping criterion $\|\mathbf{w}_{t+1} - \mathbf{w}_t\| \leq \theta = 0.01$, i.e., the next hyperplane does only a slight change.

We report the experimental results of purity, rand index, and NMI in Table 2, compactness and DB index in Table 3, and the training time (i.e., the time for finding domain of novelty) and clustering time (i.e., the time for clustering assignment) in Table 4. For each CVI, we boldface the method that yields a better outcome, i.e., highest value for purity, rand index, NMI, and DB index and lowest value for compactness. As shown in Tables 2 and 3, our proposed SGD-LMSVC is generally comparable with other baselines in the CVIs. In particular, our proposed SGD-LMSVC is slightly better than others on purity, rand index, and NMI whereas it totally surpasses others on compactness. Moreover, our proposed SGD-LMSVC is slightly worse than SVC on DB index. Regarding the amounts of time taken for training and doing clustering assignment, our proposed SGD-LMSVC is totally superior than others. For the training time, the speedup is significant for the medium-scale or large-scale datasets including Shuttle, Musk, and Abalone. In particular, the speedup is really significant for the clustering time.

5 Conclusion

In this paper, we have proposed a fast support-based clustering method, which conjoins the advantages of SGD-based method and kernel-based method. Furthermore, we have also proposed a new strategy for clustering assignment. We validate our proposed method on 15 well-known datasets for clustering task. The experiment has shown that our proposed method has achieved a comparable clustering quality compared with the baselines while being significantly faster than them.

References

1. Ben-Hur, A., Horn, D., Siegelmann, H.T., Vapnik, V.: Support vector clustering. J. Mach. Learn. Res. **2**, 125–137 (2001)
2. Camastra, F., Verri, A.: A novel kernel method for clustering. IEEE Trans. Pattern Anal. Mach. Intell. **27**(5), 801–804 (2005)
3. Chang, C.-C., Lin, C.-J.: Libsvm: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), 27:1–27:27 (2011)
4. Chu, C.S., Tsang, I.W., Kwok, J.T.: Scaling up support vector data description by using core-sets. In: Proceedings of the 2004 IEEE international joint conference on neural networks, IEEE 2004. vol. 1 (2004)
5. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Clustering validity checking methods: Part II. SIGMOD Rec. **31**(3), 19–27 (2002)
6. Hazan, E., Kale, S.: Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. J. Mach. Learn. Res. **15**(1), 2489–2512 (2014)
7. Joachims, T.: Advances in kernel methods. In: Schölkopf, B., Burges, C., Smola, A. (eds.) Making Large-Scale Support Vector

Machine Learning Practical, pp. 169–184. The MIT Press, Cambridge (1999)
8. Jung, K.-H., Lee, D., Lee, J.: Fast support-based clustering method for large-scale problems. Pattern Recognit. **43**(5), 1975–1983 (2010)
9. Le, T., Tran, D., Ma, W., Sharma, D.: An optimal sphere and two large margins approach for novelty detection. In: The 2010 international joint conference on neural networks (IJCNN), IEEE, pp. 1–6 (2010)
10. Le, T., Tran, D., Nguyen, P., Ma, W., Sharma, D.: Proximity multisphere support vector clustering. Neural Comput. Appl. **22**(7–8), 1309–1319 (2013)
11. Lee, J., Lee, D.: An improved cluster labeling method for support vector clustering. IEEE Trans. Pattern Anal. Mach. Intell. **27**(3), 461–464 (2005)
12. Lee, J., Lee, D.: Dynamic characterization of cluster structures for robust and inductive support vector clustering. IEEE Trans. Pattern Anal. Mach. Intell. **28**(11), 1869–1874 (2006)
13. Li, H.: A fast and stable cluster labeling method for support vector clustering. J. Comput. **8**(12), 3251–3256 (2013)
14. Murphy, K.P.: Machine learning: a probabilistic perspective. The MIT Press, Cambridge (2012)
15. Park, J.H., Ji, X., Zha, H., Kasturi, R.: Support vector clustering combined with spectral graph partitioning. In: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 4, pp. 581–584. IEEE (2004)
16. Pham, T., Dang, H., Le, T., Le, H-T.: Stochastic gradient descent support vector clustering. In: 2015 2nd national foundation for science and technology development conference on information and computer science (NICS), pp. 88–93 (2015)
17. Platt, J.C.: Advances in kernel methods. In: Schölkopf, B., Burges, C., Smola, A. (eds.) Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pp. 185–208. The MIT Press, Cambridge (1999)
18. Schölkopf, B., Platt, J.C., Shawe-Taylor, J.C., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Comput. **13**(7), 1443–1471 (2001)
19. Shalev-Shwartz, S., Singer, Y.: Logarithmic regret algorithms for strongly convex repeated games. The Hebrew University, Jerusalem (2007)
20. Shalev-Shwartz, S., Singer, Y., Srebro, N.: Pegasos: primal estimated sub-gradient solver for svm. In. Ghahramani, Z. (ed.) ICML, pp. 807–814 (2007)
21. Shamir, R., Sharan, R.: Algorithmic approaches to clustering gene expression data. In: Current Topics in Computational Biology. pp. 269–300. MIT Press, Cambridge (2001)
22. Tax, D.M.J., Duin, R.P.W.: Support vector data description. Mach. Learn. **54**(1), 45–66 (2004)
23. Yang, J., Estivill-Castro, V., Chalup, S.K.: Support vector clustering through proximity graph modelling. In: Neural information processing, 2002, ICONIP'02, vol. 2, pp. 898–903 (2002)